UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,567	12/30/2003	Gary L. McAlpine	10559-900001 / P17948	5728

20985 7590 11/17/2008

FISH & RICHARDSON, PC
P.O. BOX 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
CAMPOS, YAIMA	

ART UNIT	PAPER NUMBER
2185	

NOTIFICATION DATE	DELIVERY MODE
11/17/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

PATDOCTC@fr.com

Office Action Summary	Application No.	Applicant(s)	
	10/750,567	MCALPINE ET AL.	
	Examiner	Art Unit	
	YAIMA CAMPOS	2185	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 August 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9,11-24,26-44 and 46-48* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9, 11-24, 26-44, 46-48* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. As per the instant application having Application No. 10/750,567, the Examiner acknowledges Applicant's submission of the amendment dated August 8, 2008. Claims 1, 17, 26, 33, 36, 42 and 46 have been amended, and claims 10, 25, 45 and 49 have been canceled; claims 1-9, 11-24, 26-44, and 46-48 are pending in the instant Application.

REJECTIONS BASED ON PRIOR ART

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims **1-4, 8-13, 17, 20-24, 26, 33, 36, 38-42, 45-46 and 49** are rejected under 35 U.S.C. 103(a) as being unpatentable over Ikeda et al. (US 5,822,785) in view of Karkhanis et al. (US 6,085,296) and Langerman et al. (US 6,360,282).

4. As per claim 1, A machine-implemented method comprising:

receiving, by a first process in a first virtual memory address space, a shortcut to a physical address associated with a level of a multi-level virtual address translation table; **[Ikeda discloses multiprocessor communication system where multiple virtual address spaces are allocated to applications associated with the storage (col. 2, lines 40-58) where processors receive data from other processors comprising an address and a space identifier (*interpreted to*

***correspond to the claimed "shortcut" which comprises an address of a translation table*), which identifies the virtual address translation table associated with the data as either local memory space for the receiving processor or a global memory space shared by multiple processors (figs. 1, 5 and 7 and related text; col. 2, lines 51-67; col. 3, lines 8-11; col. 10, line 25-col. 11, line 20; fig. 11 and related text)]**

posting a descriptor, the descriptor comprising a virtual address in the first virtual memory address space and the shortcut, to a queue within a virtual interface that resides in a shared memory, between the first process and a second process, wherein the second process is in a second virtual memory address space that is different from the first virtual memory address space; **[Ikeda discloses space id that identifies address translation table (local or global table) information and address information is sent to transfer queue from a transmitting processing unit to a receiving processing unit wherein each processing unit is in a different virtual memory space (col. 2, lines 27-col. 4, line 33); see fig. 5 for the structure of transfer queue; note that the structure shown in fig. 1 for each processor (wherein processing devices reside in different virtual spaces) is interpreted as the virtual interface between processes as it is used for communicating data from one process in a first virtual address space to another (see. col. 2, lines 51-67; col. 3, lines 8-11; col. 10, line 25-col. 11, line 20; fig. 11 and related text)]**

notifying the second process that the descriptor has been posted to the queue via a notifying mechanism within the virtual interface **[Ikeda discloses data is transmitted from a process to another, by posting to a queue space ids and addresses (such as queue in fig. 5) wherein as**

data is received by receiving process; the receiving process is inherently notified of the transfer]

and determining, by the second process, the physical address corresponding to the virtual address associated with the first process based on at least the virtual address and the shortcut **[Ikeda discloses based on virtual address (sending and receiving) and the space id, the appropriate address translation table (local or global) is a accessed to translate virtual addresses to real addresses (col. 8, lines 4-30)]**

Ikeda does not disclose expressly that the global section which is shared by processes or applications in different virtual address spaces comprises a multi-level address translation table, and does not explicitly detail "notifying the second process that the descriptor has been posted to the queue via a notifying mechanism within the virtual interface."

Karkhanis discloses the global section which is shared by processes or applications in different virtual address spaces comprises a multi-level address translation table as **[Karkhanis discloses "the page table structures is a multi-level structure with 3 levels of page tables. Virtual address translation begins with the Page Table Base Register (PTBR) 330, which contain the physical page denoting the root 332 of a process's page table structure" (Col. 12, lines 45-59) "the data structure 400, 500 type used to manage shared leaf pages is the same as the type used to manage shared page tables 100" (Col. 4, line 56-Col. 5, line 14; Figure 1 and related text) "Global Section Descriptor (GSD) 400 describes the global section. GSD 400 includes the name 402 of the global section that is the handle used to manage the section" (Col. 13, lines 24-34) "shared page tables enable two or more processes to map to the same physical pages without each process incurring the overhead of page**

table construction… shared page tables are treated as a special case of the general management of global sections. A special global section that provides page table sharing is called a shared page table section" (Col. 8, lines 47-56) and explains "a global section is a section that can be simultaneously shared in several processes' address spaces" (Col. 6, lines 53-55)].

Langerman disclose a notifying mechanism in a virtual interface to notify that a descriptor has been posted to a queue as **[virtual interface having a queue with a send doorbell register and a receive doorbell register where a user posts I/O commands by enqueuing descriptors wherein the doorbells are used to notify of the presence of a new descriptor (col. 4, lines 25-49)]**.

Ikeda, Karkhanis and Langerman are analogous art because they are from the same field of endeavor of computer memory access and control.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to modify the system/method of transferring data between processes in different virtual memory spaces by posting transfers to a queue wherein global memory is shared among possessors and pages are translated via global page tables as taught by Ikeda, to have this page table be a multi-level page table as taught by Karkhanis, and further have the queue as taught by Ikeda include means of notifying new entries have been posted, as taught by Langerman.

The motivation for doing so would have been because Karkhanis discloses using a multi-level address translation table is used to translate entries for global memory provides the advantages of **[facilitating memory sharing among multiple processes (Col. 1, lines 54-62) and providing a shared page table so that each process does not need to incur the overhead**

of page table construction (col. 8, lines 47-56)] and Langerman discloses including a doorbell in a queue that is rung every time a new entry is added provides the advantages of **[notifying new entries have been posted to the queue (col. 4, lines 4, line 25 - col. 5, line 28)]**; thus the I/O requests would be serviced.

Therefore, it would have been obvious to combine Ikeda with Karkhanis and Langerman for the benefit of creating a method/system to obtain the invention as specified in claim 1.

5. As per claim 2, the combination of Ikeda, Karkhanis and Langerman discloses The method of claim 1 further comprising transferring data to or from a buffer located at the physical address **[Ikeda discloses bodies of data (or buffers) in different virtual address spaces are accessed when the virtual addresses are translated into physical addresses according to space id and virtual address of transfer requests (col. 13, lines 11-19; refer to fig. 11 and related text). See Karkhanis (Col. 3, lines 6-25)]**.

6. As per claim 3, the combination of Ikeda, Karkhanis and Langerman discloses The method of claim 1 further comprising: generating the shortcut by a third process **[Ikeda discloses space ids and access information is generated by any of a plurality of processes running in any of the plurality of processors in different virtual spaces in order to access/transfer data from one process to another (col. 2, lines 31-05; fig. 11 and related text). See Karkhanis (Col. 8, lines 47-56)]**.

7. As per claim 4, the combination of Ikeda, Karkhanis and Langerman discloses The method of claim 3 wherein generating the shortcut by the third process comprises: receiving a request to register a virtual buffer, the request including a virtual address corresponding to the start of the virtual buffer; determining the physical address of one level of the multi-level address

translation table associated with the virtual memory space in which the virtual buffer resides; and generating a shortcut based on the physical address of the one level of the multi-level address translation table **[Ikeda discloses bodies of data (or buffers) in different virtual address spaces are accessed when the virtual addresses are translated into physical addresses according to space id and virtual address of transfer requests (col. 13, lines 11-19; refer to fig. 11 and related text) wherein a plurality of processing units running any number of processes where data is transferred among applications running in the different processors, in different virtual spaces (col. 2, lines 31-05); thus any process may post space ids and access information to any other process, including virtual addresses to transfer queues]**.

8. As per claims 8, 21 and 38 the combination of Ikeda, Karkhanis and Langerman discloses The method of claim 1 further comprising determining if the physical address is associated with the virtual address **[Ikeda discloses (col. 12, lines 26-col. 13, line 19)]**.

9. As per claims 9, 22 and 39, the combination of Ikeda, Karkhanis and Langerman discloses The method of claim 1 further comprising determining if the virtual page containing the virtual address is pinned into physical memory **[Ikeda discloses determining whether the virtual address and space id are trying to access memory local to a processor wherein access to local memory is inhibited/protected from processes of other processors (col. 13, lines 60-67). Karkhanis discloses [memory protection and reservation; (Col. 15, lines 8-24)]**.

10. As per claims 11, 23 and 40 the combination of Ikeda, Karkhanis and Langerman discloses The method of claim 1 further comprising determining if the first process is authorized to access the virtual address, based in part, on a protection table maintained by the second process **[Ikeda discloses local and global translation and protection tables wherein global**

space is shared among processors, but access to local space of each processor is limited to the programs allowed access to the local space wherein read and write authorization tables are disclosed in figs. 13-16 (col. 13, lines 49-67; figs. 11, 13 and 18 and related text). See "Security Considerations" in Karkhanis (Col. 12, lines 16-32; Col. 10, lines 65-67 and Col. 11, lines 1-4)].

11. As per claim 12, 24 and 41 the combination of Ikeda, Karkhanis and Langerman discloses The method of claim 1 further comprising determining if descriptors posted to the interface between the first process and second process are authorized to access the virtual address, based in part, on a protection table maintained by the second process **[Ikeda discloses local and global translation and protection tables wherein global space is shared among processors, but access to local space of each processor is limited to the programs allowed access to the local space wherein read and write authorization tables are disclosed in figs. 13-16 (col. 13, lines 49-67; figs. 11, 13 and 18 and related text). See "Security Considerations" in Karkhanis (Col. 12, lines 16-32; Col. 10, lines 65-67 and Col. 11, lines 1-4)]**..

12. As per claim 13, the combination of Ikeda, Karkhanis and Langerman discloses The method of claim 1 further comprising: receiving, by a first process, a plurality of shortcuts, each shortcut to a physical address associated with a level of a multi-level virtual address translation table **[Ikeda discloses each access/transfer request transmitted to queue comprises a space identifier used as an address to access the appropriate translation table (local or global) (see. col. 2, lines 51-67; col. 3, lines 8-11; col. 10, line 25-col. 11, line 20; fig. 11 and related text), but does not disclose expressly that the translations tables include multiple levels;**

however, Karkhanis discloses multi-level address translation table used to translate entries of global shared memory, wherein this table is shared among different processes and indexed accordingly to perform address translations (Col. 12, lines 45-59); thus it would be obvious to one having ordinary skill in the art to modify the address translation tables as taught by Ikeda to be multi-level address translation tables].

13. As per claim 17, A machine-implemented method comprising: generating, by a first process in a first virtual memory address space, a request to register a virtual buffer and posting a descriptor to a queue within a virtual interface that resides in a shared memory, wherein the virtual buffer is in the first virtual memory address space and is mapped to physical memory by a multi-level virtual address translation table associated with the first process; identifying a block of memory that includes the physical address corresponding to the start of the virtual buffer; generating, by a second process, one or more shortcuts that map the block of memory that includes the physical address corresponding to the start of the virtual buffer; notifying a third process in a second virtual memory address space that is different from the first virtual memory address space that the descriptor has been posted to the queue via a notifying mechanism within the virtual interface; and transmitting a request to the third process to perform an input or output operation on the virtual buffer, wherein the request includes the shortcut and a virtual address associated with the virtual buffer **[The rationale in the rejection to claim 1 is herein incorporated. Further, Ikeda discloses bodies of data (or buffers) in different virtual address spaces are accessed when the virtual addresses are translated into physical addresses according to space id and virtual address of transfer requests (col. 13, lines 11-19; refer to fig. 11 and related text)]**.

14. As per claim 20, the combination of Ikeda, Karkhanis and Langerman discloses The method of claim 17 further comprising determining the physical address of the virtual address based on the virtual address and the shortcut **[The rationale in the rejection to claim 1 is herein incorporated]**.

15. As per claim 26, the combination of Ikea, Karkhanis and Langerman discloses A system comprising: a first processor configured to: execute instructions of a first process which causes the first processor to produce a shortcut to a physical address associated with a level of a multi-level virtual address translation table, and execute instructions of a second process in a first virtual memory address space which causes the first processor to post a descriptor comprising a virtual address and the shortcut to a queue within a virtual interface that resides in a shared memory; and notify a third process in a second virtual memory address space that is different from the first virtual memory address space that the descriptor has been posted to the queue via a notifying mechanism within the virtual interface; and a second processor configured to execute instructions of the third process in a second virtual memory address space which cause the second processor to: read the descriptor posted to the queue virtual interface, and determine a physical address of the virtual address based on at least the virtual address and the shortcut, wherein the virtual interface is between the second process and the third process and the first virtual memory address space is different from the second virtual memory address space **[The rationale in the rejection to claim 1 is herein incorporated. Further, Ikeda discloses a plurality of processing units running any number of processes where data is transferred among applications running in the different processors, in different virtual spaces (col. 2,**

lines 31-05); thus any process may post space ids and access information to any other process, including virtual addresses to transfer queues].

16. As per claim 33, the combination of Ikeda, Karkhanis and Langerman discloses A computer program product tangibly embodied in a machine-readable storage device, the computer program product comprising instructions operable to cause one or more data processing apparatus to perform operations comprising: receiving, by a first process in a first virtual memory address space, a shortcut to a physical address associated with a level of a multi-level virtual address translation table; posting a descriptor, the descriptor comprising a virtual address in the first virtual memory address space and the shortcut, to a queue within a virtual interface that resides in a shared memory, between the first process and a second process, wherein the second process is in a second virtual memory address space that is different from the first virtual memory address space; notifying the second process that the descriptor has been posted to the queue via a notifying mechanism within the virtual interface; and determining, by the second process, the physical address corresponding to the virtual address associated with the first process based on at least the virtual address and the shortcut **[The rationale in the rejection to claim 1 is herein incorporated]**.

17. As per claim 36, the combination of Ikeda, Karkhanis and Langerman discloses A computer program product tangibly embodied in a machine- readable storage device, the computer program product comprising instructions operable to cause one or more data processing apparatus to perform operations comprising: generating, by a first process in a first virtual memory address space, a request to register a virtual buffer and posting a descriptor to a queue within a virtual interface that resides in a shared memory, wherein the virtual buffer is in

the first virtual memory address space and is mapped to physical memory by a multi-level virtual address translation table associated with the first process; identifying a block of memory that includes the physical address corresponding to the start of the virtual buffer; generating, by a second process, one or more shortcuts that map the block of memory that includes the physical address corresponding to the start of the virtual buffer; notifying a third process in a second virtual memory address space that is different from the first virtual memory address space that the descriptor has been posted to the queue via a notifying mechanism within the virtual interface; and transmitting a request to the third process to perform an input or output operation on the virtual buffer, wherein the request includes the shortcut and a virtual address associated with the virtual buffer **[The rationale in the rejection to claims 1 and 17 is herein incorporated]**.

18. As per claims 42 and 46, the combination of Ikeda, Karkhanis and Langerman discloses A system comprising: a client computer; and a server in communication with the client computer using a network, the server comprising: a first processor configured to produce a shortcut to a physical address associated with a level of a multi-level virtual address translation table; and post a descriptor comprising a virtual address in a first virtual memory address space and the shortcut to a queue within a virtual interface that resides in a shared memory; and notify a third process in a second virtual memory address space that is different from the first virtual memory address space that the descriptor has been posted to the queue via a notifying mechanism within the virtual interface; and a second processor configured to perform operations in the second virtual memory address space, the operations including reading the descriptor posted to the queue, determining a physical address of the virtual address based on at least the virtual address and the

shortcut, and transferring data located at the physical address to the client computer using the network **[The rationale in the rejection to claim 1 is herein incorporated. Further, Ikeda discloses a plurality of processing units running any number of processes where data is transferred among applications running in the different processors, in different virtual spaces (col. 2, lines 31-05); thus any process may post space ids and access information to any other process, including virtual addresses to transfer queues wherein any processor may act as a server to other processors making access requests, for example to global shared memory portions. In addition, Karkhanis explains the invention is Applicable in VLM database applications supporting server processes wherein processes share global shared memory (Col. 2, line 58-Col. 3, line 5)]**.

19. **Claim 5-7, 14-16, 18-19, 27-32, 34-35, 37, 43-44 and 47-48** is rejected under 35 U.S.C. 103(a) as being unpatentable under Ikeda et al. (US 5,822,785) in view of Karkhanis et al. (US 6,085,296) and Langerman et al. (US 6,360,282) as applied to claims **1-4, 8-13, 17, 20-24, 26, 33, 36, 38-42, 45-46 and 49** above and further in view of Arndt (US 2003/0204648).

20. As per **claims 5-7, 14-16, 18-19, 25, 27-28, 34-35, 37, 43-44 and 47-48**, the combination of Ikeda, Karkhanis and Langerman discloses the method of claims 1, 4, 17, 26, 33, 36, 42 and 46 **[See rejection to claims 1, 4, 17, 26, 33, 36, 42 and 46 above]** wherein processes have different virtual memory address spaces but does not disclose expressly using a function/key to encrypt "the shortcut" which is unknown to a first/client process wherein only the host/second process is capable of decrypting said key.

Arndt discloses using function/key to encrypt "the shortcut" which is unknown to a first/client process wherein only the host/second process is capable of decrypting said key as **["a**

method, apparatus, and program for sharing logical resources among separate partitions in a logically partitioned data processing system" (Page 1, Paragraph 0002) wherein an "opaque handle" refers to an entity "which cannot be directly de-reference by the untrusted agents" thereby protecting shared resources against "untrusted agents" (Pages 4, Paragraphs 0036-0038 and Figure 3). Arndt also explains that "the hosting (client) partition uses the hypervisor function, called H_PUT RTCE, which takes as a parameter the opaque handle of the RTCE (Remote Translation Control Entry) table, such as RTCE table 330" (Page 4, Paragraph 0036) wherein only the client partition has access to the RTCE table but not the TCE (Translation Control Entry) which belongs to the host partition and maps to physical addresses. The client partition is provided an opaque handle to perform I/O operations within the host partition's memory space; therefore, preventing the client partition from containing references to a physical address of a logical resource that belongs to the host partition (Page 4, Paragraph 0042; Page 5, Paragraph 0046)].

Ikeda, Karkhanis, Langerman and Arndt are analogous art because they are from the same field of endeavor of computer memory access and control.

At the time of the invention it would have been obvious to a person of ordinary skill in the art to modify the method and system for address translation having an interface shared by different user applications and further provide determining taught by the combination of Ikeda, Karkhanis and Lagerman and use a function/key to encrypt "the shortcut" which is unknown to a first/client process wherein only the host/second process is capable of decrypting said key as taught by Arndt.

The motivation for doing so would have been because Arndt discloses use a function/key to encrypt "the shortcut" which is unknown to a first/client process wherein only the host/second process is capable of decrypting said key provides **[protection for memory resources as "by resistant to forging, the opaque handle has the characteristic such that an untrusted agent is unlikely to be able to generate, by itself, a value that would be interpreted by the hypervisor as a valid opaque handle to a TCE table" wherein the hypervisor could tell if "some other agent was trying to forge a handle to a TCE table" (Page 4, Paragraph 0038)]**.

Therefore, it would have been obvious to combine Ikeda with Karkhanis, Langerman and Arndt for the benefit of creating a method/system to control I/O requests to shared storage devices to obtain the invention as specified in claims **5-7, 14-16, 18-19, 25, 27-28, 34-35, 37, 43-44 and 47-48**.

21. As per **claim 29**, the combination of Ikeda, Karkhanis, Langerman and Arndt discloses The system of claim 28 **[See rejection to claim 28 above]** wherein the instructions of the third process cause the second processor to determine if the physical address is associated with the second process" **[The rationale in the rejection to claim 8 is herein incorporated]**.

22. As per **claim 30**, The system of claim 28 **[See rejection to claim 28 above]** "wherein the instructions of the third process cause the second processor to determine if the associated virtual pages associate with the physical address are pinned into physical memory" **[The rationale in the rejection to claim 9 is herein incorporated]**.

23. As per **claim 31**, the combination of Ikeda, Karkhanis, Langerman and Arndt discloses The system of claim 28 **[See rejection to claim 28 above]** "wherein the instructions of the third

process cause the second processor to determine if the second process is authorized to access the virtual buffer" **[The rationale in the rejection to claims 11 and 12 are herein incorporated]**.

24. As per **claim 32**, the combination of Ikeda, Karkhanis, Langerman and Arndt discloses "The system of claim 27" **[See rejection to claim 27 above]** "wherein the instructions of the third process cause the second processor to determine if requests posted to the interface between the second process and the third process are to authorized access the virtual buffer" **[The rationale to claims 11 and 12 are herein incorpoarated]**.

ACKNOWLEDGMENT OF ISSUES RAISED BY THE APPLICANT

Response to Amendment

25. Applicant's arguments filed on February 29, 2008 have been fully considered, but they are moot in view of new grounds of rejection.

CLOSING COMMENTS

Conclusion

26. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Examiner's Note

27. Examiner has cited particular columns and line numbers in the references as applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings in the art and are applied to the specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant, in preparing the responses, to fully consider the references in entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the examiner.

STATUS OF CLAIMS IN THE APPLICATION

28. The following is a summary of the treatment and status of all claims in the application as recommended by **M.P.E.P. § 707.07(i)**:

a(1) CLAIMS REJECTED IN THE APPLICATION

29. Per the instant office action, **claims 1-9, 11-24, 26-44, and 46-48** have received an action on the merits and are subject to a final rejection.

a(2) CLAIMS NO LONGER UNDER CONSIDERATION

30. Claims 10, 25, 45 and 49 have been canceled as of amendment received on August 8, 2008.

31. For at least the above reasons it is the examiner's position that the applicant's claims are not in condition for allowance.

DIRECTION OF ALL FUTURE REMARKS

32. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Yaima Campos whose telephone number is (571) 272-1232. The examiner can normally be reached on Monday to Friday 8:30 AM to 5:00 PM.

IMPORTANT NOTE

33. If attempts to reach the above noted Examiner by telephone are unsuccessful, the Examiner's supervisor, Mr. Sanjiv Shah, can be reached at the following telephone number: Area Code (571) 272-4098.

34. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

November 7, 2008

/Yaima Campos/
Examiner, Art Unit 2185

/Sanjiv Shah/

Supervisory Patent Examiner, Art Unit 2185